metraTec
RFID Solutions

MPS-Master Protocol Guide

MPS - Multi-purpose Positioning System

Date: February 2015

Version 1.1
Generic Edition

Table of Contents

1. Communication Principle 4

1.1. Helpful Tools 4

2. Instructions 5

2.1. Link (**LNK**) 6

2.2. Unlink (**ULK**) 7

2.3. Identify (**IDT**) 8

2.4. Get Power Descriptor (**PWR**) 8

2.5. Get User Descriptor (**USR**) 9

2.6. Send Configuration Data (**CFG**) 10

2.7. Set System ID (**SSID**) 11

2.8. Read System ID (**RSID**) 12

2.9. Read Extended ID (**REID**) 12

2.10. Set RSSI Threshold (**SRT**) 13

2.11. Read RSSI Threshold (**RRT**) 14

2.12. Set RF Power Level (**SPL**) 14

2.13. Read RF power level (**RPL**) 15

2.14. Set Broadcast Channel (**SCH**) 15

2.15. Read Broadcast Channel (**RCH**) 16

2.16. Set Communication Channel (**SCC**) 17

2.17. Read Communication Channel (**RCC**) 17

2.18. Set Beacon Timeout (**SBTO**) 18

2.19. Read Beacon Timeout (**RBTO**) 19

2.20. Set Data Timeout (**SDTO**) 19

2.21. Read Data Timeout (**RDTO**) 20

2.22. Set Enable Beacon (**SEBC**) 20

2.23. Read Enable Beacon (**REBC**) 21

2.24. Set Enable Data Request (**SEDR**) 22

2.25. Read Enable Data Request (**REDR**) 22

2.26. Set Enable Data (**SEDA**) 23

2.27. Read Enable Data (**REDA**) 24

2.28. Read Firmware Revision (**RFW**) 24

2.29. Read Hardwrae Revision (**RHW**) 25

2.30. Read Serial Number (**RSN**) 25

2.31. Set Heartbeat Interval (**SHI**) 26

2.32. Read Heartbeat Interval (**RHI**) 27

2.33. Reset (**RST**) 27

2.34. Factory Reset (**FRST**) 28

2.35. Set Command Lock (**SLOK**) 28

2.36. Clear Command Lock (**CLOK**) 29

2.37. Read Command Lock State (**RLOK**) 30

3. Error Codes 31

Chapter 1. Communication Principle

The communication between the device and the host system is based on ASCII strings. Each string is terminated with a carriage-return (0x0D), *not* the null byte, and will be transmitted with the most significant byte first.

The communication from the device to the host system (i.e. the response) is the same as above but in some cases the response from the device comprises more than one line.

General syntax:

```
{ Instruction } [ <SPACE> Parameter ...] <CR>
```

```
REV<CR>
```

Example 1. Command without Parameter

```
char REV[] = {'R','E','V',13};
```

Example 2. Command without Parameter in ANSI C

```
SET<SPACE>HEX<SPACE>ON<CR>
```

Example 3. Command with Parameter

```
char SET[] = {'S','E','T',0x20,'H','E','X',0x20,'O','N',0x0D};
```

Example 4. Command with Parameter in ANSI C

1.1. Helpful Tools

For debugging purpose it is very helpful to use a program to "sniff" the communication between the host and the reader. Depending on the type of communication and hardware you use, this could be:

- Since the "Keyboard TCP Bridge" supports only TCP/IP-based communication a packet sniffing tool, e.g. Wireshark/Ethereal [http://www.wireshark.org/], which is available for almost every platform
- To send ASCII data via a serial connection or even Ethernet, you can use the free metraTerm terminal software, also available on our website.

Chapter 2. Instructions

This list gives an overview of all the instructions supported by the device.

Note

Responses to any command are always prefixed with this command to avoid confusion with asynchronous messages.

```
REV<CR>
REV MPS_MASTER      0305<CR>
```

Example 5. Prefixed response

Command	Name	Description
LNK	Link	Link to a remote device.
ULK	Unlink	Terminates the link to a remote device.
IDT	Identify	Identify a remote device.
PWR	Get Power Descriptor	Get the power descriptor of a remote device.
USR	Get User Descriptor	Get the user descriptor of a remote device.
CFG	Send Configuration Data	Send configuration data to a linked device.
SSID	Set System ID	Set the Sytem ID.
RSID	Read System ID	Get the System ID.
REID	Read Extended ID	Get the Extended ID (EID).
SRT	Set RSSI Threshold	Set the RSSI threshold in dBm.
RRT	Read RSSI Threshold	Read RSSI threshold.
SPL	Set RF Power Level	Set the RF transmitter power level in dBm.
RPL	Read RF power level	Read RF power level.
SCH	Set Broadcast Channel	Set the channel for broadcast messages.
RCH	Read Broadcast Channel	Read the channel for broadcast messages.
SCC	Set Communication Channel	Set the channel for unicast communication messages.
RCC	Read Communication Channel	Read the channel for unicast communication messages.
SBTO	Set Beacon Timeout	Set the timeout for non-linked messages.
RBTO	Read Beacon Timeout	Read the timeout for non-linked messages.
SDTO	Set Data Timeout	Set the timeout for linked messages.
RDTO	Read Data Timeout	Read the timeout for linked messages.
SEBC	Set Enable Beacon	Enable or disable the display of beacon frames.
REBC	Read Enable Beacon	Read the state of beacon frame display.
SEDR	Set Enable Data Request	Enable or disable the display of data request frames.
REDR	Read Enable Data Request	Read the state of data request frame display.
SEDA	Set Enable Data	Enable or disable the display of data frames.

Command	Name	Description
REDA	Read Enable Data	Read the state of data frame display.
RFW	Read Firmware Revision	Read firmware name and version.
RHW	Read Hardwrae Revision	Read hardware name and version.
RSN	Read Serial Number	Read serial number of the device.
SHI	Set Heartbeat Interval	Set the heartbeat interval.
RHI	Read Heartbeat Interval	Read heartbeat interval.
RST	Reset	Reset the device.
FRST	Factory Reset	Reset all parameters to default values.
SLOK	Set Command Lock	Set the command lock.
CLOK	Clear Command Lock	Clear the command lock.
RLOK	Read Command Lock State	Get the command lock state.

Table 1. Overview of Instructions

2.1. Link (LNK)

Send a link request to a certain remote device with a given *Extended ID (EID)*. After the remote device responds to a link request, both devices change from the broadcast channel to the communication channel.

A link can be terminated intentionally by using the *Unlink* command (see Section 2.2, "Unlink (**ULK**)"). In case the connection is cut unintentionally a timeout exception (TOE) will occur. Possible reasons could be:

- The linked device got out of range
- The linked device was reseted or switched of
- The physical channel was disturbed by other transmitters

Note

It is not possible to have links to more than one remote device at a time.

Further on no beacon frames (BCN) will be received in linked state. Instead the periodically sent data requests (DRQ) of the linked device are displayed with the RSSI value.

```
DRQ -045<CR>
```

Example 6. Data request of a linked device with RSSI value in dBm

Instruction

LNK *<SPACE>* *{EID}* *<CR>*

Parameters

Name	Type	Description
EID	Hexadecimal String (8 bytes)	Unique Extended ID (EID) of the remote device.

Examples

```
LNK 0123456789ABCDEF<CR>
```

Example 7. Link to the remote device (EID=0123456789ABCDEF).

Return Values in Case of Success

```
LNK OK <CR>
```

Return Values in Case of Failure

```
LNK ERR <CR>
```

The last transmission hasn't finished yet. So the newer message is rejected.

“`LNK TOE <CR>`”, “`LNK UPA <CR>`” or “`LNK ERR <CR>`”

2.2. Unlink (ULK)

Send an unlink request to the linked remote device. After the remote device responds to an unlink request, both devices change back to the broadcast channel.

Instruction

```
ULK <CR>
```

Examples

```
ULK<CR>
```

Example 8. Terminate the link to a remote device.

Return Values in Case of Success

“`ULK OK <CR>`” or “`ULK TOE <CR>`”

Return Values in Case of Failure

```
ULK ERR <CR>
```

No link established to be terminated.

“`ULK UPA <CR>`” or “`ULK ERR <CR>`”

2.3. Identify (IDT)

Send an Identify request to a certain remote device with a given Extended ID (EID) if not in linked state or to the linked remote device. The remote device will start a device dependent identification sequence (e.g. flashing an LED, beeping ect.). By this a certain device can be distinguished physically from other devices very easily.

Instruction

```
IDT <SPACE> {EID} <CR>
```

Parameters

Name	Type	Description
EID	Hexadecimal String (8 bytes)	Unique Extended ID (EID) of the remote device. This parameter is optional, if you are linked to the target device.

Examples

```
IDT 0123456789ABCDEF<CR>
```

Example 9. Identify the remote device (EID=0123456789ABCDEF).

```
IDT<CR>
```

Example 10. Identify the linked remote device.

Return Values in Case of Success

```
IDT OK <CR>
```

Return Values in Case of Failure

`IDT ERR <CR>`

The last transmission hasn't finished yet. So the newer message is rejected.

“`IDT TOE <CR>`”, “`IDT UPA <CR>`” or “`IDT ERR <CR>`”

2.4. Get Power Descriptor (PWR)

Request the power descriptor from a certain remote device with a given Extended ID (EID) if not in linked state or to the linked remote device.

Instruction

```
PWR <SPACE> {EID} <CR>
```

Parameters

Name	Type	Description
EID	Hexadecimal String (8 bytes)	Unique Extended ID (EID) of the remote device. This parameter is optional, if you are linked to the target device.

Examples

```
PWR 0123456789ABCDEF<CR>
```

Example 11. Get the remote device's power descriptor (EID=0123456789ABCDEF).

```
PWR<CR>
```

Example 12. Get the linked remote device's power descriptor.

Return Values in Case of Success

`PWR {Power Descriptor}`

A 4-digit hexadecimal value containing the current power state of the remote device. Each digit contains 4 bit of information.

- 1st digit: Battery level in steps of 10%. (0: 0%, 5: 50%, A: 100%)
- 2nd digit: Bit mask of the current power source. (1: Mains, 2: Rechargeable battery, 4: Disposable battery)
- 3rd digit: Bit mask of available power sources. (1: Mains, 2: Rechargeable battery, 4: Disposable battery and binary combinations)
- 4th digit: Power mode of the receiver. (0: Always enabled, 1: Periodically enabled, 2: Enabled on external stimulus)

Return Values in Case of Failure

`PWR ERR <CR>`

The last transmission hasn't finished yet. So the newer message is rejected.

“`PWR TOE <CR>`”, “`PWR UPA <CR>`” or “`PWR ERR <CR>`”

2.5. Get User Descriptor (USR)

Request the user descriptor from a certain remote device with a given Extended ID (EID) if not in linked state or to the linked remote device.

The user descriptor contains the firmware revision of the remote device. (See Section 2.28, “Read Firmware Revision (**RFW**)”)

Instruction

USR *<SPACE>* {*EID*} *<CR>*

Parameters

Name	Type	Description
EID	Hexadecimal String (8 bytes)	Unique Extended ID (EID) of the remote device. This parameter is optional, if you are linked to the target device.

Examples

```
USR 0123456789ABCDEF<CR>
```

Example 13. Get the remote device's power descriptor (EID=0123456789ABCDEF).

```
USR<CR>
```

Example 14. Get the linked remote device's power descriptor.

Return Values in Case of Success

`USR` *{User Descriptor}*
    Firmware revision of the remote device.

Return Values in Case of Failure

`USR ERR` *<CR>*
    The last transmission hasn't finished yet. So the newer message is rejected.

"`USR TOE` *<CR>*", "`USR UPA` *<CR>*" or "`USR ERR` *<CR>*"

2.6. Send Configuration Data (CFG)

Send ASCII data to a linked device on the logical configuration channel. Following data is parsed for commands on the linked device. By this command a remote configuration of the linked device is possible.

Available commands and responses on the remote device are dependend on the certain device type and functionality and can be found in the remote device's protocol guide.

Note

It is not possible to send more than one data packet at a time. So wait for one data transmission to be finished before starting the next one.

Instruction

CFG *<SPACE>* {…*Payload*…} *<CR>*

Parameters

Name	Type	Description
Payload	Any String	Configuration data to be transmitted.

Examples

```
CFG RSN<CR>
```

Example 15. Read the remote device's serial number.

Return Values in Case of Success

`CFG  <CR>`

The resopnse in case of success depends on the remote device's protocol.

Return Values in Case of Failure

`CFG ERR <CR>`

The last transmission hasn't finished yet. So the newer message is rejected.

“`CFG TOE <CR>`”, “`CFG UPA <CR>`” or “`CFG ERR <CR>`”

2.7. Set System ID (SSID)

Configure device's System ID (SID). This setting *persists* a power up reset.

Caution

A device is only capable to communicate with other devices configured to the same System ID.

The default System ID is ABCD.

Instruction

```
SSID <SPACE> {SID} <CR>
```

Parameters

Name	Type	Description
SID	Hexadecimal Integer ($1_{16} \le x \le FFFE_{16}$)	System ID as 16 bit hexadecimal value.

Examples

```
SSID ABCD<CR>
```

Example 16. Set SID to ABCD

Return Values in Case of Success

`SSID OK <CR>`

Return Values in Case of Failure

"`SSID NOR <CR>`", "`SSID LOK <CR>`", "`SSID UPA <CR>`" or "`SSID ERR <CR>`"

2.8. Read System ID (RSID)

Returns the device's System ID (SID) (see Section 2.7, "Set System ID (**SSID**)").

Instruction

RSID *<CR>*

Examples

```
RSID<CR>
```

Example 17. Get the System ID

Return Values in Case of Success

`RSID` *{SID}*

The currently configured System ID (a 16 bit hexadecimal value).

Return Values in Case of Failure

"`RSID UPA <CR>`" or "`RSID ERR <CR>`"

2.9. Read Extended ID (REID)

Returns the device's unique Extended ID (EID).

Note

The Extended ID is unique address for every device. It is hard-coded to the device and can not be changed by the user.

Instruction

REID *<CR>*

Examples

```
REID<CR>
```

Example 18. Get the Extended ID

Return Values in Case of Success

REID *{EID}*

The currently configured System ID (a 16 bit hexadecimal value).

Return Values in Case of Failure

"REID UPA *<CR>*" or "REID ERR *<CR>*"

2.10. Set RSSI Threshold (SRT)

Configure the RSSI threshold for the reception of any wireless data in dBm. This setting *persists* a power up reset.

The default RSSI threshold is -70 dBm.

Instruction

SRT *<SPACE>* *{RSSI Threshold}* *<CR>*

Parameters

Name	Type	Description
RSSI Threshold	Decimal Integer (≤ x ≤ 0)	RSSI Threshold in *dBm*.

Examples

```
SRT -40<CR>
```

Example 19. Set RSSI Threshold to -40 dBm

```
SRT -127<CR>
```

Example 20. Set RSSI Threshold to -127 dBm

Return Values in Case of Success

SRT OK *<CR>*

Return Values in Case of Failure

"SRT NOR *<CR>*", "SRT LOK *<CR>*", "SRT UPA *<CR>*" or "SRT ERR *<CR>*"

2.11. Read RSSI Threshold (RRT)

Returns the currently configured RSSI threshold (see Section 2.10, “Set RSSI Threshold (**SRT**)”).

Instruction

RRT *<CR>*

Examples

```
RRT<CR>
```

Example 21. Read RSSI threshold

Return Values in Case of Success

RRT *{RSSI Threshold}*
The currently configured RSSI threshold (a signed decimal integer in dBm).

Return Values in Case of Failure

“RRT UPA *<CR>*” or “RRT ERR *<CR>*”

2.12. Set RF Power Level (SPL)

Configure the RF power level for the transmission of any wireless data in dBm. This setting *persists* a power up reset.

The default power level is +4 dBm.

Instruction

SPL *<SPACE>* *{RF power level}* *<CR>*

Parameters

Name	Type	Description
RF power level	Decimal Integer ($\leq x \leq 4$)	RF power level in *dBm*.

Examples

```
SPL 0<CR>
```

Example 22. Set RF power level to 0 dBm

Return Values in Case of Success

SPL OK *<CR>*

Return Values in Case of Failure

"`SPL NOR` ***<CR>***", "`SPL LOK` ***<CR>***", "`SPL UPA` ***<CR>***" or "`SPL ERR` ***<CR>***"

2.13. Read RF power level (RPL)

Returns the currently configured RF power level (see Section 2.12, "Set RF Power Level (**SPL**)").

Instruction

`RPL` ***<CR>***

Examples

```
RPL<CR>
```

Example 23. Read RF power level

Return Values in Case of Success

`RPL` *`{RF power level}`*
: The currently configured RF power level (a signed decimal integer in dBm).

Return Values in Case of Failure

"`RPL UPA` ***<CR>***" or "`RPL ERR` ***<CR>***"

2.14. Set Broadcast Channel (SCH)

Configure the channel for broadcast messages like beacon frames (BCN). This setting *persists* a power up reset.

The default broadcast channel is 26.

Note

In order to avoid collision of data frames and broadcast frames the communication channel is supposed to be different from the broadcast channel! (See Section 2.16, "Set Communication Channel (**SCC**)")

Caution

A device is only capable to communicate with other devices configured to the same broadcast channel.

Instruction

```
SCH <SPACE> {Channel} <CR>
```

Parameters

Name	Type	Description
Channel	Decimal Integer (11 ≤ x ≤ 26)	Channel number.

Examples

```
SCH 26<CR>
```

Example 24. Set channel to 26

Return Values in Case of Success

```
SCH OK <CR>
```

Return Values in Case of Failure

"SCH NOR <CR>", "SCH LOK <CR>", "SCH UPA <CR>" or "SCH ERR <CR>"

2.15. Read Broadcast Channel (RCH)

Returns the currently configured broadcast channel (see Section 2.14, "Set Broadcast Channel (**SCH**)").

Instruction

```
RCH <CR>
```

Examples

```
RCH<CR>
```

Example 25. Read broadcast channel

Return Values in Case of Success

RCH *{Channel}*

The currently configured broadcast channel number.

Return Values in Case of Failure

"RCH UPA <CR>" or "RCH ERR <CR>"

2.16. Set Communication Channel (SCC)

Configure the channel for unicast communication messages in linked state. This setting *persists* a power up reset.

The default communication channel is 25.

Note

In order to avoid collision of data frames and broadcast frames the communication channel is supposed to be different from the broadcast channel! (See Section 2.14, "Set Broadcast Channel (**SCH**)")

Instruction

SCC *<SPACE>* *{Channel}* *<CR>*

Parameters

Name	Type	Description
Channel	Decimal Integer (11 ≤ x ≤ 26)	Channel number.

Examples

```
SCC 25<CR>
```

Example 26. Set communication channel to 25

Return Values in Case of Success

`SCC OK` *<CR>*

Return Values in Case of Failure

"`SCC NOR` *<CR>*", "`SCC LOK` *<CR>*", "`SCC UPA` *<CR>*" or "`SCC ERR` *<CR>*"

2.17. Read Communication Channel (RCC)

Returns the currently configured unicast communication channel (see Section 2.16, "Set Communication Channel (**SCC**)").

Instruction

RCC *<CR>*

Examples

```
RCC<CR>
```

Example 27. Read communication channel

Return Values in Case of Success

`RCC {Channel}`
The currently configured communication channel number.

Return Values in Case of Failure

"`RCC UPA <CR>`" or "`RCC ERR <CR>`"

2.18. Set Beacon Timeout (SBTO)

Set the timeout for non-linked messages sent to a remote device. This setting *persists* a power up reset.

The default beacon timeout is 11000 milliseconds.

Note

The beacon timeout is supposed to be greater than the remote device's *Ping Interval*. A good tradeoff is a timeout twice as long as the Ping Interval.

Instruction

SBTO *<SPACE>* {*Timeout*} *<CR>*

Parameters

Name	Type	Description
Timeout	Decimal Integer (100 ≤ x ≤ 60000)	Timeout in ms.

Examples

```
SBTO 5000<CR>
```

Example 28. Set beacon timeout to 5 seconds

Return Values in Case of Success

`SBTO OK <CR>`

Return Values in Case of Failure

"`SBTO NOR <CR>`", "`SBTO LOK <CR>`", "`SBTO UPA <CR>`" or "`SBTO ERR <CR>`"

2.19. Read Beacon Timeout (RBTO)

Returns the timeout for non-linked messages sent to a remote device (see Section 2.18, “Set Beacon Timeout (**SBTO**)”).

Instruction

```
RBTO <CR>
```

Examples

```
RBTO<CR>
```

Example 29. Read beacon timeout

Return Values in Case of Success

`RBTO {Timeout}`
The currently configured beacon timeout in millisecons.

Return Values in Case of Failure

“`RBTO UPA <CR>`” or “`RBTO ERR <CR>`”

2.20. Set Data Timeout (SDTO)

Set the timeout for messages sent to a linked remote device. This setting *persists* a power up reset.

The default data timeout is 2100 milliseconds.

Note

The data timeout is supposed to be greater than the remote device's *Data Request Interval*. A good tradeoff is a timeout twice as long as the Data Request Interval.

Instruction

```
SDTO <SPACE> {Timeout} <CR>
```

Parameters

Name	Type	Description
Timeout	Decimal Integer (100 ≤ x ≤ 60000)	Timeout in ms.

Examples

```
SDTO 2000<CR>
```

Example 30. Set data timeout to 2 seconds

Return Values in Case of Success

```
SDTO OK <CR>
```

Return Values in Case of Failure

"SDTO NOR <CR>", "SDTO LOK <CR>", "SDTO UPA <CR>" or "SDTO ERR <CR>"

2.21. Read Data Timeout (RDTO)

Returns the timeout for messages sent to a linked remote device (see Section 2.20, "Set Data Timeout (**SDTO**)").

Instruction

```
RDTO <CR>
```

Examples

```
RDTO<CR>
```

Example 31. Read data timeout

Return Values in Case of Success

`RDTO {Timeout}`

The currently configured data timeout in millisecons.

Return Values in Case of Failure

"RDTO UPA <CR>" or "RDTO ERR <CR>"

2.22. Set Enable Beacon (SEBC)

Enable or disable the display of beacon frames (BCN). This setting *persists* a power up reset.

By default the display of beacon frames is enabled (1).

Note

This command only effects the display of beacon frames. Regardless of the state of this command the frames are received and processed internally for communication.

Instruction

```
SEBC <SPACE> { 0 | 1 } <CR>
```

Parameters

Name	Type	Description
State	Enumeration (0 or 1)	0: Disable BCN Display. 1: Enable BCN Display.

Examples

```
SEBC 1<CR>
```

Example 32. Enable BCN Display

```
SEBC 0<CR>
```

Example 33. Disable BCN Display

Return Values in Case of Success

`SEBC OK <CR>`

Return Values in Case of Failure

"`SEBC NOR <CR>`", "`SEBC LOK <CR>`", "`SEBC UPA <CR>`" or "`SEBC ERR <CR>`"

2.23. Read Enable Beacon (REBC)

Returns the state of beacon frame display (see Section 2.22, "Set Enable Beacon (**SEBC**)").

Instruction

REBC *<CR>*

Examples

```
REBC<CR>
```

Example 34. Read state of beacon display

Return Values in Case of Success

`REBC {State}`

The currently configured state of beacon frame display.

Return Values in Case of Failure

"`REBC UPA <CR>`" or "`REBC ERR <CR>`"

2.24. Set Enable Data Request (SEDR)

Enable or disable the display of data request frames (DRQ). This setting *persists* a power up reset.

By default the display of data request frames is enabled (1).

Note

This command only effects the display of data request frames. Regardless of the state of this command the frames are received and processed internally for communication.

Instruction

```
SEDR <SPACE> { 0 | 1 } <CR>
```

Parameters

Name	Type	Description
State	Enumeration (`0` or 1)	0: Disable DRQ Display. 1: Enable DRQ Display.

Examples

```
SEDR 1<CR>
```

Example 35. Enable DRQ Display

```
SEDR 0<CR>
```

Example 36. Disable DRQ Display

Return Values in Case of Success

```
SEDR OK <CR>
```

Return Values in Case of Failure

"`SEDR NOR <CR>`", "`SEDR LOK <CR>`", "`SEDR UPA <CR>`" or "`SEDR ERR <CR>`"

2.25. Read Enable Data Request (REDR)

Returns the state of data request frame display (see Section 2.24, "Set Enable Data Request (**SEDR**)").

Instruction

```
REDR <CR>
```

Examples

```
REDR<CR>
```

Example 37. Read state of data request display

Return Values in Case of Success

REDR *{State}*

The currently configured state of data request frame display.

Return Values in Case of Failure

"`REDR UPA <CR>`" or "`REDR ERR <CR>`"

2.26. Set Enable Data (SEDA)

Enable or disable the display of data frames (DAT). This setting *persists* a power up reset.

By default the display of data frames is enabled (1).

Note

This command only effects the display of data frames. Regardless of the state of this command the frames are received and processed internally for communication.

Instruction

```
SEDA <SPACE> { 0 | 1 } <CR>
```

Parameters

Name	Type	Description
State	Enumeration (0 or 1)	0: Disable DAT Display. 1: Enable DAT Display.

Examples

```
SEDA 1<CR>
```

Example 38. Enable DAT Display

```
SEDA 0<CR>
```

Example 39. Disable DAT Display

Return Values in Case of Success

```
SEDA OK <CR>
```

Return Values in Case of Failure

"`SEDA NOR` *`<CR>`*", "`SEDA LOK` *`<CR>`*", "`SEDA UPA` *`<CR>`*" or "`SEDA ERR` *`<CR>`*"

2.27. Read Enable Data (REDA)

Returns the state of data frame display (see Section 2.26, "Set Enable Data (**`SEDA`**)").

Instruction

`REDA` *`<CR>`*

Examples

```
REDA<CR>
```

Example 40. Read state of data display

Return Values in Case of Success

`REDA` *`{State}`*

The currently configured state of data frame display.

Return Values in Case of Failure

"`REDA UPA` *`<CR>`*" or "`REDA ERR` *`<CR>`*"

2.28. Read Firmware Revision (RFW)

Returns the firmware name and version.

Instruction

`RFW` *`<CR>`*

Examples

```
RFW<CR>
```

Example 41. Read firmware revision

Return Values in Case of Success

`RFW` *`{Name}{Major}{Minor}`*

The name of the firmware is padded to 16 characters with spaces, followed by a two-digit major revision, followed by a two-digit minor revision.

For instance version 3.5 of MPS_MASTER firmware yields the following response to `RFW`:

```
RFW MPS_MASTER      0305<CR>
```

Example 42. Revision Response

Return Values in Case of Failure

"`RFW UPA` *<CR>*" or "`RFW ERR` *<CR>*"

2.29. Read Hardwrae Revision (RHW)

Returns the hardware name and version.

Instruction

RHW *<CR>*

Examples

```
RHW<CR>
```

Example 43. Read hardware revision

Return Values in Case of Success

`RHW` *{Name}{Major}{Minor}*

The name of the hardware is padded to 16 characters with spaces, followed by a two-digit major revision, followed by a two-digit minor revision.

For instance version 1.2 of ERS_PoE hardware yields the following response to `RHW`:

```
RHW ERS_PoE         0102<CR>
```

Example 44. Revision Response

Return Values in Case of Failure

"`RHW UPA` *<CR>*" or "`RHW ERR` *<CR>*"

2.30. Read Serial Number (RSN)

Returns the serial number of the device.

Instruction

RSN *<CR>*

Examples

```
RSN<CR>
```

Example 45. Read serial number

Return Values in Case of Success

`RSN {Serial Number}`
The serial number consists of 16 characters coding the date, time and place of production.

```
RSN 2014103119391200<CR>
```

Example 46. Serial Number Response

Return Values in Case of Failure

"`RSN UPA <CR>`" or "`RSN ERR <CR>`"

2.31. Set Heartbeat Interval (`SHI`)

Configure heart beat interval. This setting *persists* a power up reset.

The application asynchronously sends heartbeat messages in the interval configured with **`SHI`**. A heart beat message looks like:

```
HBT<CR>
```

Example 47. Heartbeat Message

By default the heartbeat function is disabled.

Instruction

`SHI` `<SPACE> {Interval} <CR>`

Parameters

Name	Type	Description
Interval	Decimal Integer (x ≥ 0)	Heartbeat interval in *seconds*. 0 disables heartbeat messages.

Examples

```
SHI 20<CR>
```

Example 48. Set heartbeat interval to 20 seconds

```
SHI 0<CR>
```

Example 49. Disable heartbeat messages

Return Values in Case of Success

`SHI OK! <CR>`

Return Values in Case of Failure

"`SHI NOR <CR>`", "`SHI LOK <CR>`", "`SHI UPA <CR>`" or "`SHI ERR <CR>`"

2.32. Read Heartbeat Interval (RHI)

Returns the currently configured heartbeat interval (see Section 2.31, "Set Heartbeat Interval (**SHI**)").

Instruction

RHI *<CR>*

Examples

```
RHI<CR>
```

Example 50. Read heartbeat interval

Return Values in Case of Success

`RHI` *{Interval}*

The currently configured heartbeat interval (an unsigned decimal integer in seconds).

Return Values in Case of Failure

"`RHI UPA <CR>`" or "`RHI ERR <CR>`"

2.33. Reset (RST)

Performs a hardware reset.

Instruction

RST *<CR>*

Examples

```
RST<CR>
```

Example 51. Read serial number

Return Values in Case of Success

`RST OK <CR>`

Return Values in Case of Failure

"`RST LOK <CR>`", "`RST UPA <CR>`" or "`RST ERR <CR>`"

2.34. Factory Reset (FRST)

Resets all parameters to default values and performs a new initialization afterwards.

Instruction

FRST *<CR>*

Examples

```
FRST<CR>
```

Example 52. Reset all parameters to default values.

Return Values in Case of Success

`FRST OK <CR>`

Return Values in Case of Failure

"`FRST LOK <CR>`", "`FRST UPA <CR>`" or "`FRST ERR <CR>`"

2.35. Set Command Lock (SLOK)

Enables the command lock and sets the lock password. The lock affects *all set-commands*, *RST*, *FRST* and *BTL*. This setting *persists* a power up reset.

Caution

To release the lock you will need the previously set password. (See Section 2.36, "Clear Command Lock (**CLOK**)")

If you forgot the password it is imposible to release the command lock!

Instruction

SLOK *<SPACE>* {*Password*} *<CR>*

Parameters

Name	Type	Description
Password	Hexadecimal Integer ($0_{16} \leq x \leq FFFFFFFF_{16}$)	Arbitrarily chosen password as 32 bit hexadecimal value.

Examples

```
SLOK 01234567<CR>
```

Example 53. Set lock with password 01234567

Return Values in Case of Success

SLOK OK *<CR>*

Return Values in Case of Failure

"SLOK NOR *<CR>*", "SLOK LOK *<CR>*", "SLOK UPA *<CR>*" or "SLOK ERR *<CR>*"

2.36. Clear Command Lock (CLOK)

Releases the command lock with the previously set password. (See Section 2.35, "Set Command Lock (**SLOK**)") This setting *persists* a power up reset.

Instruction

CLOK *<SPACE>* {*Password*} *<CR>*

Parameters

Name	Type	Description
Password	Hexadecimal Integer ($0_{16} \leq x \leq FFFFFFFF_{16}$)	Password as 32 bit hexadecimal value.

Examples

```
CLOK 01234567<CR>
```

Example 54. Release lock with password 01234567

Return Values in Case of Success

`CLOK OK <CR>`

Return Values in Case of Failure

“`CLOK NOR <CR>`”, “`CLOK WPW <CR>`”, “`CLOK UPA <CR>`” or “`CLOK ERR <CR>`”

2.37. Read Command Lock State (RLOK)

Returns the command lock state. (See Section 2.35, “Set Command Lock (**SLOK**)”)

Instruction

`RLOK` `<CR>`

Examples

```
RLOK<CR>
```

Example 55. Get the command lock state

Return Values in Case of Success

`RLOK {State}`
The currently configured state of the command lock.

Return Values in Case of Failure

“`RLOK UPA <CR>`” or “`RLOK ERR <CR>`”

Chapter 3. Error Codes

Error Code	Name	Description
`ERR`	Unknown Error	An unknown (miscellaneous) error occurred.
`LOK`	Command locked	The requested command was locked. (See Section 2.35, “Set Command Lock (**`SLOK`**)”)
`NOR`	Number Out of Range	This error code may refer to an input parameter or output value. When referring to an input parameter, it signifies that a parameter is too small or large. On output, it means that the number cannot be represented — usually because of a numeric overflow.
`NOS`	Not Supported	Command or parameter not supported by this specific device type
`TOE`	Timeout Exception	A timeout exception occured during communication to a remote device.
`UCO`	Unknown Command	An invalid command has been passed to a function. Common error source: • Typo in command string • Wrong firmware version
`UPA`	Unknown Parameter	An invalid parameter has been passed to a function. Common error source: • Typo in command string • Given parameter is out of range • Parameter missing (formerly EPX)
`WPW`	Wrong password	The entered password is not correct.

Version Control

Version	Change	By	Date
1.0	created	DT	31.10.2014
1.1	Response prefixes added. SPL and RPL commands added.	DT	02.02.2015

metraTec GmbH

Werner-Heisenberg-Str. 1
39106 Magdeburg
Germany

Tel.: +49 (0)391 251906-00
Fax: +49 (0)391 251906-01

Email: <support@metratec.com>
Web: http://www.metratec.com